



Introduktion till ingenjörsarbete inom datateknik, DT502G

Grundläggande programmering, teori
1,5 högskolepoäng

Skriftlig tentamen

2018-11-20

Introduktion till ingenjörarbete inom datateknik, DT502G

Tillåtna hjälpmedel: penna, radergummi.

Instruktioner:

- Läs igenom alla frågor noga.
- Skriv bara på ena sidan av svarsbladet.
- Skriv tentamenskoden på varje svarsblad.
- Du kan svara på *Svenska* eller *Engelska*.
- *Skriv läsligt!*

Ansvarig lärare: Amy Loutfi, tel: 0722149853, och Andreas Persson, tel: 019-303714 (alt. 0707760861)

För betyg G krävs 50% av total poäng.

Lycka till!

Part I: Open answers questions

Answer each question on a separate page.

Question 1

The following function is used for calculating the element-wise division between the elements of two lists of integers:

```
def listDivision(x, y):  
    z = []  
    for i in range(len(x)):  
        z.append(int(x[i] / y[i]))  
    return z
```

```
>>> a = [1, 8, 9, 5, 16, 24, 0, 8]  
>>> b = [1, 2, 3, 0, 4, 6, 5]  
>>> c = listDivision(a, b)  
>>> print(c)
```

However, there are *two* major problems that cause errors (and raises two different types of *exceptions*) when this function is used in the example below. *State those problems. (2p)*

Question 2

A *string* object has a number of methods for operating on the object (e.g. `upper`, `lower`, `split`, etc.). Unfortunately, there exists no such method for returning the *reversed string*.

Your task is, therefore, to write a *function* that accepts an *arbitrary string* as the *parameter*, uses a `for` loop to build the *reverse string*, and *returns* the reversed string, according to the example below. **(2p)**

```
>>> s = 'hello world'  
>>> r = reverse(s)  
>>> print(r)  
dlrow olleh
```

Question 3

Repeat **Question 2** but this time using a `while` loop instead. **(2p)**

Question 4

The following function is intended to be used for calculating the *least common denominator* (LCD) for the two integer numbers n_1 and n_2 :

```
def LCD(n1, n2):
    if n1 == n2:
        return n1
    elif n1 > n2:
        d = n1
        while (d % n2) > 0:
            d = d + n1
        return d
    else:
        d = n2
        while (d % n1) > 0:
            d = d + n2
        return d
```

However, the function has lost its *indentation* and it will (obviously) not work as intended. Fix the indentation so that the function works as intended. **(2p)**

Tip, use a graph paper (sv. rutat papper) to show your answer.

Question 5

Analyze the following function and statement:

```
def func(lst, n):
    for i in range(2, n):
        if n % i == 0:
            return
    lst.append(n)

lst = []
for i in range(10, 20):
    func(lst, i)
print(lst)
```

What will be the print output of the statement above? *Justify your answer.* **(2p)**

Question 6

Write a `factorial` function that accepts a non-negative integer n as the parameter and ***recursively*** calculates the product of all positive integers less than or equal to n . **(2p)**

For example, in case $n = 5$, the function should return: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

Part II: Multiple choice questions

Answer by selecting the option(s) that you think best matches the question.

Question 7

What will the output be for the following `for` loop? (1p)

```
for i in range(1, 10, 2):  
    print(i, end=' ')
```

Note, the "end" parameter is simply used to
override the default newline at the end of
the print statement.

- a) 0 1 2 3 4 5 6 7 8 9
- b) 1 2 3 4 5 6 7 8 9
- c) 1 3 5 7 9
- d) 0 2 4 6 8

Question 8

What is the right condition to ensure that a float number `x` is within the interval: `[0.0, 1.0]`? (1p)

- a) `x >= 0.0 and x <= 1.0`
- b) `x >= 0.0 or x <= 1.0`
- c) `x in [0.0, 1.0]`
- d) `x > 0.0 or x < 1.0`

Question 9

Which of the following Python statements will **not** give an answer of 3.0? (1p)

- a) `>>> 27 / 3 / 3`
- b) `>>> math.sqrt(9)`
- c) `>>> 9 ** (1/2)`
- d) `>>> 27 % 3`

Question 10

Consider the following function:

```
def splitStr(x, y, z):  
    txt = 'pythons are snakes'  
    words = txt.split(' ')  
    print(words[z][y] + words[y][z] + words[x][z])
```

What is the print output for the function call: `splitStr(-1, 1, -3)`? (1p)

- a) ska b) yak c) nes d) yes

Question 11

Consider the following class definition:

```
class ComplexNumber:  
  
    def __init__(x, r = 0, i = 0):  
        x.real = r  
        x.imag = i  
  
    def addReal(x, r):  
        x.real += r  
  
    def addImag(x, i):  
        x.imag += i  
  
    def __str__(x):  
        return "{0} + {1}j".format(x.real, x.imag)
```

A: What is the *keyword* used instead of the `x` (used in the class definition above) to indicate that a method or variable is a member of the class? (1p)

- a) this
b) def
c) self
d) class

B: Given the class above, what will be the output for the following example? (1p)

```
c1 = ComplexNumber(2, 3)  
c1.addReal(2)  
c2 = c1  
c2.addImag(1)  
print(c1)
```

- a) 2 + 3j
b) 4 + 3j
c) 4 + 4j
d) *Nothing, there will be an error instead.*

Question 12

What is **not true** about `if-elif-else` statements? (1p)

- a) An `if` must always be accompanied by an `else`
- b) An `elif` must always be accompanied by an initial `if`
- c) An `else` can never have a *condition*.
- d) Both an `if` and an `elif` can have several *conditions*.

Question 13

Given the following *dictionary* and statement:

```
>>> X = {'A': 1, 'C': 2, 'B': 3, 4: 'D', 'E': 5}
>>> for k in X.keys():
...     print(X[k], end=' ')
```

What will be the print output of statement above? (1p)

- a) 1 2 3 4 5
- b) 1 2 3 D 5
- c) A C B D E
- d) A C B 4 E